ELSEVIER
Discrete Applied Mathematics 74 (1997) 1–12
DISCRETE APPLIED MATHEMATICS

Finding the maximum and minimum

Martin Aigner

Freie Universitaet Berlin, II. Mathematisches Institut, Arnimallee 3, D-14195 Berlin, Germany

Received 15 September 1995

Abstract

We consider the problem of finding the maximum out of a list of n ordered items with binary comparisons where the pth fraction of the answers may be false. It is shown that the maximum can be determined iff $p < \frac{1}{2}$ and that a successful strategy needs $\Theta(\frac{1}{1-p})^n$ questions. A few similar problems are also discussed, including the problem of finding the maximum and minimum simultaneously with lies and in the nuts and bolts model.

Keywords: Sorting; Searching; Erroneous information; Threshold

1. Introduction

Several papers have recently appeared on search problems when some of the answers are lies. Two models have received particular attention. In the first model a fixed number k of the answers may be false (see e.g. [4–6, 8–10]), whereas in the second model a fixed proportion p of the answers may be erroneous (see [3, 7, 11]). In the present paper we consider first the problem of finding the maximum of a list of n distinct elements where the tests are binary comparisons. The case of a fixed number $l-1$ of lies was solved in [8]: The worst-case search length is $ln - 1$. We treat the second model. As popularized by Spencer we consider the search process as a game between two players Paul (asking the questions) and Carole (providing the answers). Paul wins if he can determine the maximum, otherwise Carole wins.

Our main result proved in the next section is the following

Theorem 1. *Suppose* $0 < p \leqslant 1$, *and the list contains* $n \geqslant 3$ *elements whose maximum is to be determined. At each stage of the game Carole may have given false answers to at most the pth fraction of the questions asked up to this point. If*

(a) $p \geqslant \frac{1}{2}$, *then Carole wins,*
(b) $p < \frac{1}{2}$, *then Paul wins and he has to ask* $\Theta(\frac{1}{1-p})^n$ *questions.*

Note that this is in marked contrast to the membership problem where $p = \frac{1}{2}$ is again the threshold value, but Paul has to ask only $O(\log n)$ questions (see [11]).

SSDI 0166-218X(96)00012-1

The method of proof will also establish the threshold $p = \frac{1}{2}$ for *any* sorting problem. In Section 3 we consider some related questions and discuss in Section 4 the problem of determining simultaneously the maximum and minimum in the presence of errors. In the last section the worst-case cost of finding the maximum resp. maximum and minimum is determined in the nuts and bolts model introduced by Alon et al. [2].

2. Proof of Theorem 1

Suppose first $p \geqslant \frac{1}{2}$. We adopt the usual jargon, calling a comparison $x : y$ a *game*, and saying x *wins* resp. y *loses* if $x > y$. Carole answers according to a poset until only two candidates (maximal elements) a and b are left. That is, she provides true answers which are compatible with all previous relations. Since every $x \neq a, b$ must have lost at least once, it takes Paul at least $n-2$ games to reduce the number of candidates to two. From now on Carole answers alternately $a < b$, $a > b$, $a < b, \ldots$ and gives true answers $x < a$, $x < b$ otherwise. Hence after $(n-2)+(2l-1)$ questions, Carole has lied at most l times, and since $l \leqslant \frac{1}{2}(n-2+2l-1)$ for $n \geqslant 3$, Paul never knows which of a or b is the true maximum.

Assume now $p < \frac{1}{2}$. Paul adopts the strategy of comparing the current maximum with a new element so often until one of them drops out. At the start he compares $a : b$. The answer must be true, so the loser, say b, drops out. Now Paul compares $a : c$. Again the loser drops out since $p < \frac{1}{2}$. Assume inductively that it takes $c_1, c_2, \ldots, c_{i-1}$ games ($i \geqslant 3$) to eliminate the 1st, $\ldots$, $(i-1)$th element, and set $S_{i-1} = \sum_{j=1}^{i-1} c_j$. Thus $c_1 = c_2 = 1$ and $S_{n-1} =$ total number of games. Let x be the current maximum, and suppose y is an element not yet considered. Paul compares $x : y$ a number of times. Suppose there are a_i outcomes $x < y$ and b_i outcomes $y < x$, where we assume w.l.o.g. $a_i \geqslant b_i$, $c_i = a_i + b_i$. The element x must drop out as soon as

$$\frac{a_i + b_{i-1} + \cdots + b_1}{c_i + S_{i-1}} > p, \tag{1}$$

where b_j is the number of lies encountered in the jth round ($b_1 = b_2 = 0$), since $x < y$ must then be the true answer. From $a_i \geqslant \frac{c_i}{2}$ we conclude

$$\frac{a_i + b_{i-1} + \cdots + b_1}{c_i + S_{i-1}} \geqslant \frac{a_i}{2a_i + S_{i-1}} = \frac{1}{2 + S_{i-1}/a_i},$$

and this last expression is $> p$ for a_i large enough. Hence Paul wins by induction. We show next that Paul needs at most $O(\frac{1}{1-p})^n$ questions. We use the notation a_j, b_j, c_j, S_j as before. Hence b_j is the number of lies occurring in the jth round (eliminating the jth element). The key to the proof is the observation that Carole does best if she never lies until only two candidates are left, i.e. $b_j = 0$ for $j \leqslant n-2$. Suppose Carole lies first in the ith round ($i \leqslant n-2$), say b_i times. We show that if she lies once less in round i and once more in round $i+1$, then S_{i+1} is at least as large as before. Repeating this argument will prove the observation.

Since a_i is the smallest number satisfying (1) we have

$$\frac{a_i - 1}{a_i + b_i - 1 + S_{i-1}} \leqslant p, \tag{2}$$

hence

$$(1 - p)a_i \leqslant p(b_i + S_{i-1}) + 1 - p. \tag{3}$$

Now suppose Carole lies only $b_i - 1$ times, and let $a_i - d$ be the smallest number satisfying (1). Then

$$\frac{a_i - d}{a_i - d + b_i - 1 + S_{i-1}} > p,$$

hence

$$(1 - p)d < (1 - p)a_i - p(b_i + S_{i-1}) + p.$$

By (3) this yields $(1 - p)d < 1$, and thus $d \leqslant 1$ since $p < \frac{1}{2}$. So Carole reduces a_i by $d \leqslant 1$ and c_i by $d + 1 \leqslant 2$. It remains to show that c_{i+1} increases by at least $d + 1$. Setting $a'_{i+1} = a_{i+1} + d$, $b'_{i+1} = b_{i+1} + 1$, we must verify that $a'_{i+1} - 1$ does not satisfy (1) with $b'_i = b_i - 1$, $S'_i = S_i - d - 1$. Since a_{i+1} is the smallest integer satisfying (1) for $i + 1$, we have

$$a_{i+1} - 1 + b_i \leqslant p(a_{i+1} - 1 + b_{i+1} + S_i). \tag{4}$$

Hence we conclude

$$\begin{aligned} a'_{i+1} - 1 + b'_i = a_{i+1} + d + b_i - 2 &\leqslant a_{i+1} - 1 + b_i \\ &\leqslant p(a_{i+1} - 1 + b_{i+1} + S_i) \\ &= p(a'_{i+1} - d - 1 + b'_{i+1} - 1 + S'_i + d + 1) \\ &= p(a'_{i+1} - 1 + b'_{i+1} + S'_i), \end{aligned}$$

and this is precisely what we wanted to show. Hence we may assume $b_j = 0$ for $j \leqslant n - 2$. Let $i \leqslant n - 2$. We infer from (2) with $a_i = c_i$

$$c_i - 1 \leqslant p(c_i - 1 + S_{i-1}),$$

and thus

$$(1 - p)c_i \leqslant pS_{i-1} + 1 - p.$$

Adding $(1 - p)S_{i-1}$ to both sides yields

$$S_i \leqslant \frac{1}{1 - p}S_{i-1} + 1 \quad (i \leqslant n - 2). \tag{5}$$

In the last round, the best Carole can do is to answer alternately, and we obtain $a_{n-1} = \frac{1}{2}(c_{n-1} + 1)$, and thus with (2)

$$\tfrac{1}{2}(c_{n-1} - 1) \leqslant p(c_{n-1} - 1 + S_{n-2}),$$

$$(1-2p)c_{n-1} \leqslant 2pS_{n-2} + 1 - 2p,$$

$$S_{n-1} \leqslant \frac{1}{1-2p}S_{n-2} + 1.$$

Hence $S_{n-1} = \mathrm{O}(\frac{1}{1-p})^n$ follows by (5).

It remains to show that Carole has a counterstrategy forcing Paul to ask at least $\Omega(\frac{1}{1-p})^n$ questions. Carole gives true answers according to a fixed linear order. Suppose the ith element x_i is eliminated in game d_i, $d_1 < d_1 < \cdots < d_{n-1}$. Now if there is a game with x_{i+1} as loser before game d_i (which has x_i as loser), interchange these two games. This does not change the chances of x_j ($j \leqslant i-1$) or x_{i+1} being eliminated and increases the chances of x_i. Hence this change does not favor Carole, and we may assume that all games involving x_i as loser come before all games with x_{i+1} as loser ($1 \leqslant i \leqslant n-2$).

Let c_i be the number of losses of x_i, and set $S_i = \sum_{j=1}^{i} c_j$. To eliminate x_i we must have

$$\frac{c_i}{c_i + S_{i-1}} > p,$$

since all previous games have true answers. We conclude

$$(1-p)c_i > pS_{i-1}, \qquad (1-p)S_i > S_{i-1}, \qquad S_i > \frac{1}{1-p}S_{i-1},$$

and thus $S_{n-1} = \Omega(\frac{1}{1-p})^n$. $\square$

Notice that the step-by-step algorithm of Paul can be applied to any sorting problem. Suppose $p < \frac{1}{2}$ and consider an arbitrary sorting procedure of length $\mathrm{O}(n \log n)$ to produce the full linear order. At any step $a : b$ of the algorithm Paul compares $a : b$ so often until the true outcome $a < b$ or $b < a$ is determined. By the same argument as in the proof of Theorem 1, the best Carole can do is to reserve all lies to the last comparison. Hence we have the following:

Corollary. *Consider the full sorting problem. If $p \geqslant \frac{1}{2}$, then Carole wins. If $p < \frac{1}{2}$, then Paul wins and he needs to ask at most $(\mathrm{O}(\frac{1}{1-p}))^{\mathrm{O}(n \log n)}$ questions.*

Since in the full sorting problem the maximum has to be determined at any rate we note the lower bound $\Omega(\frac{1}{1-p})^n$ (see in this connection [4]).

Similar results can be derived for the median problem or for selection problems (see [1]).

3. Two other models

In [11], Spencer and Winkler discuss two other models (B) and (C) for the membership problem, favoring Carole as compared to the procedure (A) considered so far.

In model (B), Carole may lie as often as she wishes but at the end she may have given false answers to at most the pth fraction. In model (C) Paul puts down his questions, and Carole may choose the pth fraction of questions to which she gives erroneous answers. Model (C) is clearly an error-correcting problem for binary codes. Spencer and Winkler proved the threshold $p = \frac{1}{2}$ for our previous situation (A), and $p = \frac{1}{3}$ resp. $p = \frac{1}{4}$ for model (B) resp. (C).

For the maximum problem considered here the answers for (B) resp. (C) are easy (and not independent of n).

Proposition 1. *Consider the problem of finding the maximum out of a list of* $n \geqslant 2$ *elements in model* (B). *If*

(a) $p \geqslant \frac{1}{n-1}$, *then Carole wins,*

(b) $p < \frac{1}{n-1}$, *then Paul wins with* $n-1$ *questions.*

Proof. If $p < \frac{1}{n-1}$, then the first $n-1$ answers must be true, and Paul wins by applying the usual procedure. Suppose $p \geqslant \frac{1}{n-1}$. Assume on the contrary that Paul wins with $l(n-1)+k$, $1 \leqslant k \leqslant n-1$, $l \geqslant 0$, games. Carole answers according to a poset.

Case 1: $1 \leqslant k \leqslant n-2$. Since $l(n-1)+k < l(n-1)+n-1 = (n-1)(l+1)$, there must be two players a, b who have lost at most l times. Since $p \geqslant \frac{1}{n-1}$, the number of possible lies is at least $l + \frac{k}{n-1} \geqslant l$. Hence if Carole lied exactly when a lost, a is a candidate. Similarly, b is still a candidate, and Paul cannot win.

Case 2: $k = n-1$. Here $l(n-1)+(n-1) = (n-1)(l+1)$. We infer that two players a, b have lost at most $l+1$ times. Since $p \geqslant \frac{1}{n-1}$, the number of possible lies is at least $l+1$, and we proceed as in case 1. □

Proposition 2. *Consider the problem of finding the maximum out of a list of* $n \geqslant 2$ *elements in model* (C). *If*

(a) $p \geqslant 1/\binom{n}{2}$, *then Carole wins,*

(b) $p < 1/\binom{n}{2}$, *then Paul wins with* $\binom{n}{2}$ *questions.*

Proof. If $p < 1/\binom{n}{2}$, then Paul puts all $\binom{n}{2}$ comparisons down, and Carole must answer truthfully.

Suppose $p \geqslant 1/\binom{n}{2}$. Let α_{xy} be the number of comparisons $x : y$, $q = \sum \alpha_{xy}$ be the total number of questions. Let u, v be a pair with $\alpha_{uv} = \min \alpha_{xy}$, then

$$\alpha_{uv} \leqslant q \Big/ \binom{n}{2} \leqslant pq, \qquad \frac{\alpha_{uv}}{q} \leqslant p.$$

Carole chooses a linear order $x_1 < \cdots < x_{n-2} < u < v$, and Paul does not know whether u or v is the maximum, since all α_{uv} answers to $u : v$ may be one way or the other. □

4. Finding the maximum and minimum

It is well known that $\lceil \frac{3n}{2} \rceil - 2$ is the search length for finding the maximum and minimum out of a list of $n \geqslant 2$ elements, when no errors are present (see e.g. [1]). Considering model (A), we conclude from the remarks in Section 2 that Paul wins precisely for $p < \frac{1}{2}$, and that he needs no more than $O(\frac{1}{1-p})^{3n/2}$ questions. Since he must at any rate determine the maximum, $\Omega(\frac{1}{1-p})^n$ is a lower bound, but the precise growth seems difficult to determine.

Let us consider the case of a fixed number $l-1$ of lies, $l \geqslant 2$. Even the situation with one lie ($l = 2$) appears quite involved. We derive below upper and lower bounds.

Lemma. *Suppose Carole may lie $l-1$ times. As long as a player has lost at most $l-1$ times, he may still be the maximum. Similarly, as long as a player has won at most $l-1$ times, he may still be the minimum.*

Proof. Carole answers according to a poset. Suppose player a has lost to $b_1, \ldots, b_t$, $t \leqslant l-1$. Since no relation $a < b_i$ has been forced by transitivity in the poset constructed so far, turning the relations $a < b_i$ around yields again a poset, and a becomes a candidate for the maximum. □

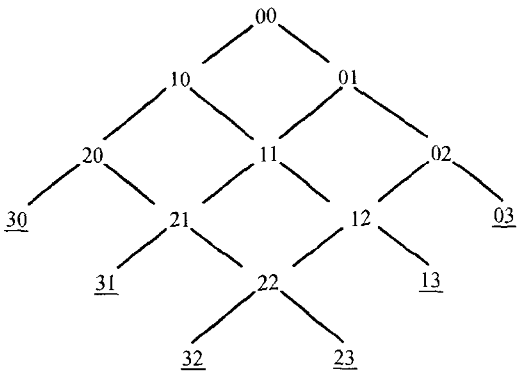
Consider a stage of the search process. We call x an (i,j)-player if x has won i times and lost j times, where for $i \geqslant l$ resp. $j \geqslant l$ we always set $i = l$ resp. $j = l$. Thus the players are partitioned into $(l+1)^2$ classes (i,j), $0 \leqslant i,j \leqslant l$. Call $i+j$ the *weight* of an (i,j)-player. According to the Lemma, only the (l,l)-players can be eliminated from further consideration. We know that the search length for the maximum is $ln-1$, so we can certainly determine the maximum and minimum with $2ln-2$ tests. The orders ln for the lower bound and $2ln$ for the upper bound are improved in the following result.

Theorem 2. *Let $L(n,l)$ be the search length for n elements and $l-1$ lies, $l \geqslant 2$. Then*

$$a_l l_n - 2 \leqslant L(n,l) \leqslant A_l l_n + C_l,$$

where $a_l = 1 + l^{-1}\sqrt{2}[(1+\sqrt{2})^{l-1} - (1-\sqrt{2})^{l-1}]^{-1}$, $A_l = 1 + \binom{2l}{l}2^{-2l}$, and C_l depends only on l.

Proof. Let us consider the upper bound first. Suppose $n = k2^{2l}$, and partition the elements into k subsets of size 2^{2l}. In every 2^{2l}-set Paul proceeds as follows. He makes 2^{2l-1} disjoint comparisons, thus producing 2^{2l-1} $(1,0)$-elements and 2^{2l-1} $(0,1)$-elements. Now he compares the $(1,0)$-elements in pairs and similarly the $(0,1)$-elements. In general, he compares the elements within an (i,j)-group in pairs until a (l,i)-group resp. (j,l)-group is produced. This first step can be conveniently represented in the following diagram. As an example, consider $l = 3$, then we obtain the diagram:



Let t_i be the number of elements in group (l, i) after this step. By symmetry, this is also the number of elements in group (i, l). Since there are $\binom{l-1+i}{i}$ lattice paths from $(0,0)$ to $(l-1, i)$, there are $\binom{l-1+i}{i}2^{l+1-i}$ elements in group $(l-1, i)$, and thus

$$t_i = \binom{l-1+i}{i}2^{l-i} \quad (i = 0, \ldots, l-1) \tag{6}$$

with

$$\sum_{i=0}^{l-1} t_i = 2^{2l-1}. \tag{7}$$

From (6) we infer

$$\sum_{i=0}^{l-1} it_i = \sum_{i=1}^{l-1} i\binom{l-1+i}{i}2^{l-i} = \sum_{i=1}^{l-1} l\binom{l-1+i}{i-1}2^{l-i}$$

$$= \frac{l}{4}\sum_{i=0}^{l-2}\binom{l+i}{i}2^{l+1-i},$$

and hence by (7)

$$\sum_{i=0}^{l-1} it_i = \frac{l}{4}\left(2^{2l+1} - \binom{2l-1}{l-1}2^2 - \binom{2l}{l}2\right).$$

Since $2^2\binom{2l-1}{l-1} = 2\binom{2l}{l}$, this yields

$$\sum_{i=0}^{l-1} it_i = l\left(2^{2l-1} - \binom{2l}{l}\right). \tag{8}$$

Let us count the number L_1 of games played in this step. The weight of every player is raised by 1 in each game in which he is involved. Thus taking into account the players of type (j,l), $j<l$, we obtain by (7) and (8)

$$L_1 = \sum_{i=0}^{l-1}(l+i)t_i = l\sum_{i=0}^{l-1} t_i + \sum_{i=0}^{l-1} it_i,$$

$$L_1 = l\left(2^{2l} - \binom{2l}{l}\right). \tag{9}$$

In the second stage, Paul compares the elements within each group (l,i) resp. (j,l), until only one remains. Since the weights are increased by only 1 in each game, we need in this stage $L_2 = 2\sum_{i=0}^{l-1}(l-i)(t_i-1)$ games. Using (7) and (8) again, we obtain

$$L_2 = 2l\binom{2l}{l} - l^2 - l. \tag{10}$$

Performing the same question procedure in every one of the k groups of size 2^{2l}, Paul has so far asked

$$\begin{aligned} L' &= k(L_1+L_2) = kl\left(2^{2l} + \tbinom{2l}{l}\right) - k(l^2+l) \\ &= ln\left(1 + \tbinom{2l}{l}2^{-2l}\right) - k(l^2+l) \end{aligned} \tag{11}$$

questions. Now, there are k elements of each type (l,i) resp. (j,l) left. Paul compares the candidates (l,i) for the maximum resp. the candidates (j,l) for the minimum. Every game raises the weight by 1, so another

$$\begin{aligned} L'' &= 2k(l+(l-1)+\cdots+1)-2 \\ &= k(l^2+l)-2 \end{aligned}$$

games are needed, and we obtain

$$L(n,l) \leqslant L' + L'' = ln\left(1 + \binom{2l}{l}2^{-2l}\right) - 2.$$

If n is not of the form $n = k2^{2l}$, take the largest number $n' \leqslant n$, $n' \equiv 0 \pmod{2^{2l}}$ and handle the remaining elements in a straightforward fashion. This proves the upper bound.

We turn to the lower bound. Let us denote by W_i the sum of the weights after i questions. Thus at the start $W_0 = 0$, whereas at the end after L questions, we have $W_L = 2ln-2$ by the Lemma. Any question increases the total weight by at most 2. We call a question "good" if it increases the weight by 2, otherwise "bad". Hence, if G and B denote the number of good resp. bad questions, then $L = G+B$, $2G+B \geqslant 2ln-2$, and thus

$$L \geqslant ln + \tfrac{B}{2} - 1. \tag{12}$$

Our goal is therefore to bound B from below. Carole answers according to a poset, thereby using the following oracle:

$$(0,l) < (0,l-1) < \cdots < (0,1) < \{(i,j) : i,j \geqslant 1\} < (1,0) < \cdots < (l,0).$$

That is, Carole answers $a < b$, whenever a is a $(0,l)$-player and b is a $(0,l-1)$-player, similarly for the other relations, giving arbitrary answers in the remaining cases, compatible with the previous relations. Let U_i $(i = 1,\dots,l)$ be the number of players who, after i questions, have type $(i,0)$ or $(0,i)$. Hence U_i counts the players who win their first i games or lose their first i games. Obviously, $U_1 = n \geqslant U_2 \geqslant \cdots \geqslant U_l$. Consider type $(j-1,0)$ or $(0,j-1)$. Suppose a is a $(j-1,0)$-player. By the setup of the oracle, a can lose in his next game only against a $(k,0)$-player b with $k \geqslant j-1$. If b is a $(j-1,0)$-player, then one of a or b becomes a $(j,0)$-player, and if b is a $(l,0)$-player, then $a : b$ is a bad game. Denoting by $C_{j-1,k}$ $(j \leqslant k \leqslant l)$ the number of games between a $(j-1,0)$-player and a $(k,0)$-player or between a $(0,j-1)$-player and a $(0,k)$-player, we infer

$$U_j \geqslant \frac{U_{j-1} - C_{j-1,j} - \cdots - C_{j-1,l}}{2} \tag{13}$$

with $C_{j-1,l}$ bad games. Rearranging terms, this yields

$$\begin{array}{lllllll}
C_{1,l} & \geqslant n - 2U_2 - C_{1,2} - C_{1,3} & - \cdots - & C_{1,l-2} & -C_{1,l-1} & & \\
C_{2,l} & \geqslant \qquad U_2 - 2U_3 - C_{2,3} & - \cdots - & C_{2,l-2} & -C_{2,l-1} & & \\
\vdots & & & & & & \\
C_{l-1,l} & \geqslant & & U_{l-1} & & - & 2U_l \\
C_{l,l} & \geqslant & & & & & lU_l - 2.
\end{array} \tag{14}$$

Since in any of the $C_{j-1,k}$ games $(j \leqslant k \leqslant l-1)$ a different $(k+1,0)$ resp. $(0,k+1)$-player is produced, we infer

$$C_{1,k} + C_{2,k} + \cdots + C_{k-1,k} \leqslant U_{k+1} \quad (1 \leqslant k \leqslant l-1) \tag{15}$$

and

$$B \geqslant C_{1,l} + \cdots + C_{l,l}. \tag{16}$$

Let us attach positive weights b_j to the jth inequality $(1 \leqslant j \leqslant l)$ in (14), so that the right-hand side consists of non-negative terms only. Set $b_1 = 1$, $b_2 = 2$ and $b_j = 2b_{j-1} + b_{j-2}$ for $2 \leqslant j \leqslant l-1$, $b_l = b_{l-1}$. We claim that on the right-hand side of $b_1C_{1,l} + \cdots + b_jC_{j,l}$ of (14) all terms up to and including the jth column have non-negative entries. For $j = 1$ we have $b_1 n = n$. Assume the assertion is correct up to $j-1$. Since the $(j-1)$th column has zeros from row j on, it suffices by induction to consider the jth column. The original entries are $-C_{1,j-1},\dots,-C_{j-2,j-1},-2U_j,U_j$. From (15) we obtain for $j \leqslant l-1$

$$-b_1C_{1,j-1} - \cdots - b_{j-2}C_{j-2,j-1} - 2b_{j-1}U_j + b_jU_j \geqslant (-b_{j-2} - 2b_{j-1} + b_j)U_j = 0,$$

and similarly for the lth column.

We conclude from (16) and (14)

$$b_{l-1}B \geqslant b_1 C_{1,l} + \cdots + b_l C_{l,l} \geqslant n - 2b_{l-1}$$

and thus $\frac{B}{2} \geqslant \frac{n}{2b_{l-1}} - 1$.

Solving the recurrence for the b_i's, we obtain

$$b_{l-1} = \frac{1}{2\sqrt{2}}((1+\sqrt{2})^{l-1} - (1-\sqrt{2})^{l-1}),$$

and the lower bound results from (12). □

Remark. For one lie ($l = 2$) we obtain $\frac{5}{2}n - 2 \leqslant L(n,2) \leqslant \frac{11}{4}n + C_2$, and it is easily shown that our procedure yields $C_2 = 0$. Both bounds can be improved. Using a refined oracle one can show $L(n,2) \geqslant \frac{8}{3}n - 2$ for $n \geqslant 3$, and the upper bound can be improved to $L(n,2) \leqslant \frac{87}{32}n$. The exact growth of $L(n,2)$ is not known, but we surmise that the upper bound is closer to the truth. In general, the upper bound in Theorem 2 appears much stronger than the lower bound provided by the oracle.

5. The nuts and bolts model

A very interesting search model was recently studied by Alon et al. [2]. Consider a set $X = \{x_1, \ldots, x_n\}$ of bolts of different sizes which are to be put in linear order. In addition, we are given a set $Y = \{y_1, \ldots, y_n\}$ of matching nuts in some order. To find the proper ordering of the bolts x_i we can only compare the sizes of a bolt x_i and a nut y_j with the outcome $x_i = y_j$, $x_i < y_j$ or $x_i > y_j$. How many comparisons are needed? In mathematical terms, we are given a set $X = \{x_1, \ldots, x_n\}$ endowed with an unknown linear order and some (unknown) permutation $\{y_1, \ldots, y_n\}$ of the x_i's, and we are only allowed comparisons $x_i : y_j$. It was shown in [2] that at most $O(n \log^{3+\varepsilon} n)$ comparisons are needed for the full sorting problem, and it is an intriguing question whether $O(n \log n)$ comparisons suffice as in the ordinary case.

For the maximum resp. maximum and minimum problem we can give a precise answer which turns out to be twice the number of comparisons in the usual case.

Theorem 3. *Let $M(n)$ be the worst-case cost for the maximum problem with n elements in the nuts and bolts model, then $M(n) = 2n - 2$.*

Proof. Clearly, $M(1) = 0$ and $M(n) = 2$. The lower bound is easily established. Let $x_1 : y_1$ be the first comparison. The oracle (Carole) answers $x_1 = y_1$ and treats x_1 as minimum. Hence by induction we need $2(n-1) - 2 = 2n - 4$ comparisons to determine the maximum among the x_i ($i \geqslant 2$) and one more to be sure that x_1 is not the maximum.

Let us turn to the upper bound. At the beginning, all $2n$ elements x_i, y_j are candidates for the maximum. Now we only compare candidates. Whenever a comparison $x_i : y_j$ has answer $x_i < y_j$ or $x_i > y_j$, then x_i or y_j drops out, reducing the number of candidates by 1. Suppose $x_i = y_j$, and let $X', Y' = \{y_1, \dots, y_t\}$ be the candidates before the comparison $x_i : y_j$. Clearly $|X'| \geqslant 2$, since otherwise $x_i : y_j$ is not necessary. After the outcome $x_i = y_j$ we compare x_i one by one with the other $y_k \in Y'$. If $x_i < y_k$ for some k, then x_i *and* y_j drop out, and if $x_i > y_k$ for all $k \neq j$, then x_i is the maximum and all of $X' - \{x_i\}$ drop out which means that in the last comparison at least two elements drop out since $|X'| \geqslant 2$. We conclude that with every answer $x_i = y_j$ there is another comparison where two candidates drop out. Hence after at most $2n - 2$ comparisons, only two candidates are left, one in X and one in Y, and the proof is complete. □

Theorem 4. *Let $MM(n)$ be the worst-case cost for determining the maximum and minimum in the nuts and bolts model, then $MM(n) = 3n - 4$.*

Proof. Trivially, $MM(2) = 2$ so let us assume $n \geqslant 3$. Let us look at the lower bound first. As in the previous proof, Carole answers $x_1 = y_1$ to the first comparison, and treats x_1 as both non-maximal and non-minimal. By induction, $3(n-1) - 4 = 3n - 7$ comparisons are necessary for the remaining $n - 1$ elements, and it is easily seen that two more are needed to eliminate x_1 for sure.

Now to the upper bound. Paul performs first the n disjoint comparisons $x_i : y_i$ $(i = 1, \dots, n)$. Suppose there are l outcomes $x_i < y_i$, $x_i > y_i$ and $n - l$ outcomes $x_i = y_i$.

Case i: $l = 0$. In this case $x_1 = y_1, \dots, x_{n-1} = y_{n-1}$ whence the last comparison $x_n = y_n$ is superfluous. Since we are now in the ordinary max–min case, we conclude

$$\text{cost} \leqslant (n-1) + \left\lceil \tfrac{3}{2}n \right\rceil - 2 = \left\lceil \tfrac{5}{2}n \right\rceil - 3 \leqslant 3n - 4 \quad \text{for } n \geqslant 3.$$

Case ii: $1 \leqslant l \leqslant n-1$, say $x_i \neq y_i$ $(i = 1, \dots, l)$. By induction, we need $2l - 4$ further comparisons to determine the top element $\overline{a}$ and bottom element $\underline{a}$ among these l pairs. Now we are in the ordinary case involving $\underline{a}, \overline{a}, x_i = y_i$ $(i = l+1, \dots, n)$ where we already know one relation $\underline{a} < \overline{a}$, and we infer

$$\begin{aligned}\text{cost} &\leqslant n + 2l - 4 + \left\lceil \tfrac{3}{2}(n - l + 2) \right\rceil - 3 \\ &\leqslant \tfrac{5}{2}n + \tfrac{1}{2}l + \tfrac{1}{2} - 4 \leqslant 3n - 4\end{aligned}$$

since $l \leqslant n - 1$.

Case iii: $l = n$. After the n initial comparisons there remain n candidates for the maximum and n candidates for the minimum. Look at the maximum candidates. We now proceed as in the proof of the upper bound in the previous theorem. After at most $n - 2$ comparisons there are only two candidates left, one in X and one in Y, and the maximum is determined. Applying the same reasoning for the minimum candidates, we conclude

$$\text{cost} \leqslant n + 2(n-2) = 3n - 4,$$

and the proof is complete. □

References

[1] M. Aigner, Combinatorial Search (Wiley-Teubner, 1988).
[2] N. Alon, M. Blum, A. Fiat, S.K. Kannan, M. Naor and R. Ostrovsky, Matching nuts and bolts, in: Proc. Fifth Annual ACM-SIAM SODA (ACM Press, New York, 1994) 690–696.
[3] J.A. Aslam and A. Dagat, Searching in the presence of linearly bounded errors, in: Proc. 23rd ACM Symp. on Theory of Computing (1991) 486–493.
[4] R.S. Borgstrom and S.R. Kosaraju, Comparison-based search in the presence of errors, STOC 1993.
[5] A. De Bonis, L. Gargano and U. Vaccaro, Group testing with unreliable tests, Inform. Process. Lett., submitted.
[6] A. Pelc, Solution of Ulam's game on searching with a lie, J. Combin. Theory (A) 44 (1987) 129–140.
[7] A. Pelc, Searching with known error probability, Theoret. Comput. Sci. 63 (1989) 185–202.
[8] B. Ravikumar, K. Ganesan and K.B. Lakshmanan, On selecting the largest element inspite of erroneous information, in: Proc. ICALP 87, 88–99.
[9] R.L. Rivest, A.R. Meyer, D.J. Kleitman, K. Winklmann and J. Spencer, Coping with errors in binary search procedures, J. Comput. System. Sci. 20 (1980) 396–404.
[10] J. Spencer, Ulam's searching game with a fixed number of lies, Theoret. Comput. Sci. 95 (1992) 307–321.
[11] J. Spencer and P. Winkler, Three thresholds for a liar, Combin. Probab. Comput. 1 (1992) 81–93.